User's Guide VISIPART

SIMONA report number 2008-01

Rijkswaterstaat
Ministerie van Infrastructuur en Milieu

User's Guide VISIPART

Version number : Version 1.3, June 2011
Maintenance : see www.helpdeskwater.nl/waqua
Copyright : Rijkswaterstaat

Log-sheet

document version	date	Changes with respect to the previous version	
1.0	16-01-2008	c77580	update of old documentation Visipart
1.1	19-10-2010	c3320	Support in Visipart for DDHOR-simulation
1.2	21-10-2010	c3320	Converted to LaTeX
1.3	10-06-2011	c3320	Note on compiled version and removed empty appendix

Contents

1 Directions for use of this manual **1**

2 Description of the menu-items of Visipart **2**

2.1 Starting and stopping Visipart 2

2.2 Options of the File menu 2

2.2.1 File | Import WAQUA enclosure 2

2.2.2 File | Apply partitioning 3

2.2.3 File | Load partitioning 3

2.2.4 File | Apply WAQUA enclosure 3

2.2.5 File | Save partitioning 4

2.2.6 File | Print 4

2.2.7 File | Exit 4

2.3 Options of the Edit menu 4

2.3.1 Edit | Undo 4

2.3.2 Edit | Set subdomain number, Left mouse button 4

2.3.3 Edit | Get subdomain number, Right mouse button 5

2.3.4 Edit | Remap 5

2.3.5 Edit | Split subdomain 5

2.3.6 Edit | Split all subdomains 6

2.4 Options of the View menu 6

2.4.1 View | Zoom in 6

2.4.2 View | Zoom out 6

2.4.3 View | Zoom Region 6

2.4.4 View | Zoom Full 6

2.4.5 View | Max Aspect Ratio 6

2.4.6 View | Show Grid 7

2.4.7 View | Show Curvilinear 7
2.5 Options of the Options menu 7
2.5.1 Options | Show estimated workload 7
2.5.2 Options | Show Fullbox 7
2.5.3 Options | Show Fill-ratio 7
2.5.4 Options | Set Number of Layers 7
2.5.5 Options | Set Computing Speeds 8
2.5.6 Options | Show Observed Run-times 8
2.5.7 Options | Assign Processors to Subdomains 9
2.5.8 Options | Check Domain Decomposition 9

3 Optimizing grid partitionings **10**

4 Example for assigning processes in DDHOR simulation using Visipart **12**

Chapter 1

Directions for use of this manual

Visipart is a Matlab-based, graphical tool to assist in creating or modifying grid partitionings or grid decompositions for use with parallel computing or domain decomposition. This document describes the use of the tool. Chapter 2 describes the functions of the program, and Chapter 3 describes a way to use the tool in order to optimize partitionings for a given simulation.

Chapter 2

Description of the menu-items of Visipart

2.1 Starting and stopping Visipart

Visipart can be started using the run-procedure visipart.pl. This will start Matlab for you, add the directory of the Visipart program files to the Matlab search-directory, and start its graphical interface. This will bring up an empty figure window with a menu-bar that is specific for Visipart. Only one instance of Visipart may run at any moment. Starting a second one results in an error message.

The perl-procedure visipart.pl is located in the bin-directory of a SIMONA installation. Visipart should work ok when the appropriate settings for the SIMONA installation have been made.

It is possible to start Visipart directly from the Matlab command prompt. For this you should add a directory to the Matlab search path as follows:
`addpath('<SIMONADIR>/matlab/visipart');`.
After this Visipart can be started by executing the command visipart at the Matlab command prompt.

On Windows, one can start Visipart by running the compiled version which can be found in <SIMONADIR>/bin/win3
. One advantage of the compiled version is the possibility to have more than one instance of Visipart running at the same time.

To exit VISIPART, select menu item File | Exit.

2.2 Options of the File menu

2.2.1 File | Import WAQUA enclosure

To start creating a new decomposition for a model geometry, select an enclosure file via this menu item. The enclosure is usually specified in the siminp-file or an include-file. You may either select the whole siminp-file to be used or select the include file that contains the computational grid enclosure. The enclosure will be read, processed, all interior points will be assigned to subdomain 1, and the resulting partitioning will be shown in the Visipart window.

2.2.2 File | Apply partitioning

When you have imported a WAQUA enclosure as described above (File | Import WAQUA enclosure), you may combine this with a partitioning that is described in a partitioning input-file. This is useful when the partitioning is given through (possibly large) rectangular boxes or enclosures that do not show the actual geometry. When the partitioning is restricted to the actual grid this allows for reading curvilinear grid coordinates too.

The partitioning-files supported here are the same as those supported by File | Load partitioning described below.

This option can be used only when there have not been created any subdomains, i.e. when all grid points of the enclosure are assigned to subdomain 1.

Grid points that are active in the enclosure (subdomain=1) and not active in the partitioning (subdomain=0) will be assigned to a separate subdomain.

2.2.3 File | Load partitioning

Select this menu-item to load an existing partitioning. This brings up a file-selection dialogue from which the partitioning file can be selected. Once the partitioning has successfully been loaded, it will be shown in the Visipart window.

All partitioning files (parallel computing) and areas files (domain decomposition) that are allowed by Coppre should be supported by Visipart too. For information on their formats consult the Users guide for Parallel WAQUA / TRIWAQ and for Domain decomposition.

Partitioning/decomposition files may be created by hand, by using Visipart (see File | Save partitioning), or by using Coppre, which writes the partitioning used to the report-file *"coppre-r.<runid>"*.

Visipart determines the size of the grid that must be partitioned from the extension of the partitioning that is stored in the file. So, if you use an areas format and specify MNMNBOXes, make sure that the extension of the boxes together match those of the grid itself. Or otherwise start by reading the WAQUA enclosure into Visipart, and after that apply the partitioning (File | Apply partitioning).

2.2.4 File | Apply WAQUA enclosure

When you have read a partitioning as described above (File | Load partitioning), it may not yet show the actual geometry of the WAQUA grid. This may be accomplished by restricting the current partitioning to the grid-points enclosed by a WAQUA computational grid enclosure. This combination operation is activated by the current menu item.

When you select this menu item a pop-up window shows up where you can select an enclosure-file. You may either select the whole siminp-file to be used or select the include file that contains the computational grid enclosure. The enclosure will be read and processed. Then the current partitioning will be adjusted. All grid points selected by the enclosure that have not been assigned to a subdomain will be assigned to a new subdomain "npart+1". Then the subdomain value is kept for interior points and reset to 0 for other points of the grid. Finally the resulting partitioning will be shown in the Visipart window.

2.2.5 File | Save partitioning

To store a partitioning, select this menu item. This brings up a file-selection dialogue in which the name of the file can be set. The file is saved in the areas format using "SUBDOMAIN"s and "ENCLOSURE"s.

2.2.6 File | Print

This menu item allows to print the current view of the partitioning using a printer or a file. For this the Matlab standard printing options are used.

2.2.7 File | Exit

This menu item is used to exit from Visipart. When you have made modifications to the partitioning since the last moment you saved the partitioning or imported a new enclosure or partitioning, a question will be asked whether you really want to exit or want to return to Visipart to save your work first.

2.3 Options of the Edit menu

2.3.1 Edit | Undo

If you make an error while modifying the partitioning, select this menu item to undo the latest modification.

2.3.2 Edit | Set subdomain number, Left mouse button

To modify a partitioning, press and hold down the left mouse button at some point in the partitioning and drag the mouse. This will show a selected area in the partitioning. Once you let go of the left mouse button, a dialogue will appear asking for the subdomain number to which the selected area must be assigned.

Typing a number that does not yet occur in the partitioning will create a new subdomain. Domains have a positive number, but for one exception. Subdomain –1 is used to identify the "inactive part" of the domain in a DDHOR simulation. This inactive part should be filled in by another domain (siminp-file). (See also the example in Section 4)

The menu item "Edit | Set subdomain number" brings up a small informational message that points you to using the left mouse button.

2.3.3 Edit | Get subdomain number, Right mouse button

The subdomain numbers are visible in the partitioning. In some cases, it may be difficult to determine to which subdomain a particular grid point belongs. Press the right mouse-button at that grid point to get the number of the subdomain to which it belongs. This is also useful for setting the default value for modifying areas with the left mouse button.

The menu item "Edit | Get subdomain number" does nothing more than point you to using the right mouse button.

2.3.4 Edit | Remap

To assign a new number to an entire subdomain, select this menu item. This will bring up a dialogue asking for the new subdomain number.

2.3.5 Edit | Split subdomain

A subdomain in the partition can easily be split up into smaller subdomains using one of the automatic partitioning methods that are also available in Coppre. The menu item "Edit | Split subdomain" is used to split one subdomain into smaller subdomains.

After selecting this menu item a pointer (cross-hair) comes up in the partitioning window which must be used to select a subdomain with the mouse. Then a window pops up in which you can configure the automatic partitioning method to be used:

- the number of subdomains that must be created;
- the partitioning method to be used (SGRB, ORB or Strip-wise);
- the intial orientation for splitting the subdomain (automatic, rowwise or column-wise).

The SGRB partitioning method is usually the best for partitioning of WAQUA grids. It combines the favourable properties of the ORB and Strip-wise methods. It consists of recursively splitting the subdomain into two parts, along the shortest dimension of each subdomain at each stage.

The ORB partitioning method consists of recursively splitting the subdomain into two parts, thereby in each stage using an orientation orthogonal to the orientation of the previous stage.

The Strip-wise partitioning method consists of splitting the subdomain into the requested number of parts using the same orientation for all subdomains.

When using the automatic orientation option, the initial orientation depends on the sizes in m- and n-directions of the initial subdomain. If the subdomain has less rows than columns, the initial split will be row-wise, assigning the first "X" rows to part 1 and the remaining "Y" rows to part 2.

The implementation supports any number of subdomains. For instance when you request 5 subdomains, the values of X and Y will be determined such that the amount of work for the first X rows is about 1.5 times the work of that for the remaining Y rows. Then the method will recursively split the first part into 3 sub-parts and the second part into 2 sub-parts, such that each part approximately gets

the same amount of work. Note that the estimated work-load may be investigated using menu-item "Options | Show workload".

2.3.6 Edit | Split all subdomains

Menu-item "Edit | Split all subdomains" is used for splitting all subdomains simultaneously into smaller subdomains, all using the same method as described under "Edit | Split subdomain" above. This option is in particular useful for creating a finer partition using an already existing (optimised) partition.

2.4 Options of the View menu

2.4.1 View | Zoom in

To get a closer look at the partitioning, select this menu item. When only a part of the partitioning is in the figure, sliders will be visible next to the partitioning. These can be used to scroll through the partitioning. The effect of the slider move will not become visible until de mouse-button is released.

2.4.2 View | Zoom out

View a larger part of the partitioning. The region that is displayed will be enlarged by approximately a factor 1.8.

2.4.3 View | Zoom Region

When you have selected this menu-item, you will be prompted to mark the area in the grid that you want to have enlarged. Push and hold down the left mouse-button at one corner of the desired area (which must be inside the figure showing the partitioning) and drag the mouse to the opposite corner. When you release the left mousebutton, the zooming will take place.

2.4.4 View | Zoom Full

Get a full view of the partitioning again.

2.4.5 View | Max Aspect Ratio

This selection will rescale the figure to its maximal size in the Visipart window. This option is not available when viewing a curvilinear grid.

2.4.6 View | Show Grid

This option will display a coordinate grid over the partitioning to help determine the coordinates of points in the partitioning. This option is only effective when viewing the grid using computational (m,n)-coordinates.

2.4.7 View | Show Curvilinear

In case of a curvilinear grid, the partitioning can be shown in physical coordinates by selecting this menu item. When selecting this option for the first time for a particular grid, you will be prompted for the name of the RGF-file in which coordinates are stored.

2.5 Options of the Options menu

2.5.1 Options | Show estimated workload

Select this menu-item for viewing an estimate of the relative amount of computing time needed per domain. This estimate is derived from the number of grid points in each subdomain times the number of layers, divided by the relative computing speed per domain, when given (see below). This is a useful feature when creating a partitioning for parallel computing, but will not often be used when selecting the subdomains for domain decomposition with vertical refinement.

2.5.2 Options | Show Fullbox

This option is meant for giving information on the characteristics of each subdomain. It shows the size of the rectangular hull around the domain, which is the dimension of arrays the computational core of Waqua/Triwaq.

2.5.3 Options | Show Fill-ratio

This option is meant for giving information on the characteristics of each subdomain. It shows the ratio of the number of active (wet) grid points over the total number of grid points (i.e. including land points) in the full box.

2.5.4 Options | Set Number of Layers

When using domain decomposition with vertical grid refinement (DDVERT), the number of layers may vary per subdomain. This is relevant for Visipart when you are combining DDVERT with parallel computing, when you want to split each DDVERT subdomain into different parts with about the same runtime per processor.

The number of layers per subdomain is used in estimating the workload per subdomain (see "Options | Show Estimated Workload" above), and in suggesting the number of sub-subdomains per subdomain in option "Options | Assign Processors to Subdomains" below.

The number of layers per subdomain are initially all equal to 1, so that the estimated workload is equal to the number of active grid points per subdomain. The actual numbers of layers in each subdomain can be entered in pop-up windows that are brought up when this option is selected. The pop-up window displays the current values of the settings for all subdomains and the default value that is used when new subdomains are created. These values can be modified by selecting an input-region and typing the desired values. The number of layers should be a positive integer value. The new values are accepted when the pop-up window is closed by pressing the OK button.

2.5.5 Options | Set Computing Speeds

For each subdomain a "relative computing speed" may be entered. This speed is used in constructing the graph with estimated workload per subdomain (see "Options | Show Estimated Workload").

One application of using varying computing speeds per subdomain is when both faster and slower computers are used in a single run. Another application is found in optimizing grid partitionings. This latter application primarily uses "Options | Show Observed Run- Times" as described below and in Chapter on optimizing grid partitionings.

Like the number of layers per subdomain, the relative computing speeds are initially equal to 1. The actual relative computing speeds are set in a way very similar to the number of layers. The relative computing speed can be any positive value larger than 1e-5.

2.5.6 Options | Show Observed Run-times

The relative computing speed per subdomain can be filled in automatically from the run-times that are observed in an actual simulation using a given partitioning of the domain. This option first brings up a file-selection dialog box in which the waqpro message file should be selected. This message file is then parsed by Visipart so it gets all the information needed. Note that all lines are evaluated one by one, so it is advised to keep the message file as short as possible. For example we advise not to use the nmdbg application during a simulation when you want to use Visipart also.

Next, another window is created in which the run-times are displayed in a graph, and estimates for the computing speed are shown in textual format. The estimates are derived by combining the total calculation times and the workload (number of grid points times number of layers) of the actual decomposition. The estimates are used in the estimated workload per domain only after the Accept speeds button is pressed. Note that good estimates are obtained only when the message file corresponds to the current partitioning in Visipart.

In case of a DDHOR-simulation the runtimes for all processors from each domain are shown. In the figure that pops up it is clearly shows which times are for which domain. When the Accept speeds button is pressed a box pops up with the question which domain is currently being edited. If an existing name is given and OK is pushed, the estimated speeds for that domain are used for the estimated workload per domain (See also Section 4).

2.5.7 Options | Assign Processors to Subdomains

This option was developed with the vertical refinement option of Triwaq in mind, particularly for the combined usage of domain decomposition and parallel computing. The idea is that the domain decomposition is created first, i.e. the regions with different numbers of layers are selected, and then the different regions are further subdivided into an appropriate number of subdomains. The option allows for computing an appropriate distribution of processors over the different regions, thereby taking into account the workload in each region (number of active grid points times number of layers).

When the option is selected, this brings up a pop-up window in which a range can be given for the number of processors that is considered. Then for all numbers of processors in this range the optimal assignment of processors to regions is determined assuming perfect parallellisation, and the resulting computing time and cost are set out in a graph.

The computing time shown is the estimated workload for the largest subdomain in a run scaled by the computing time for the run with the largest number of processors. The computing cost is computed as the computing time times the number of processors used. These values are scaled by the computing time for a single processor run. This is equal to one over the efficiency of the parallel run.

Below the graph, the optimal assignment is shown for one of the numbers of processors in the range.

When the decomposition of the global domain into regions is modified, the computed distribution of processors over the regions may become sub-optimal. In this case the Recompute button may be pressed for determining the new optimal assignments of processors and for updating the graph.

For a DDHOR-simulation this option is only available if the computing times for an actual run are available. This is because Visipart only knows about one of the domains; the measured run times provide information on the other domains. If the observed runtimes are not available to Visipart yet, Visipart asks whether this is a DDHOR-simulation or not. If you answer yes, then Visipart will kindly ask you to load the observed run-times first.

2.5.8 Options | Check Domain Decomposition

This option can be used to check whether a partitioning is a valid domain decomposition partitioning. This option is enabled only when the grid is shown in rectilinear coordinates (see also menu-item “View | Show Curvilinear”).

A partitioning is valid for domain decomposition if the subdomain interfaces are far enough from each other and from the open boundaries. When the option “Options | Check Domain Decomposition” is selected, a dialog box will pop up that reports potential errors; an arrow in the partitioning will show where the problematic point is located.

Visipart checks the border of each subdomain in turn. Because it has no information about boundary types, Visipart cannot see the difference between closed and open boundaries. Consequently, it will produce more warnings than necessary. If you are sure that a certain subdomain interface is OK, select ‘next segment’ in the dialog box to continue checking in the next part of the border of the subdomain

Chapter 3

Optimizing grid partitionings

The following approach has proven useful for the optimisation of a partitioning.

1. An initial partitioning is created in the following steps:
    a. Read the computing domain's geometry using the function **"File | Import WAQUA enclosure"**.
    b. Optionally select regions with the mouse to assign to subdomains with different numbers of layers for using vertical refinement.
    c. When using vertical refinement, set the number of layers per subdomain using **"Options | Set number of Layers"**.
    d. When using vertical refinement, distribute the available processors over the subdomains using the function **"Options | Assign processors to domains"**.
    e. Split the global domain (when not using vertical refinement) or each subdomain (when using vertical refinement) in the desired number of subdomains using **"Edit | Split subdomain"**.
    f. Save the partitioning using **"File | Save partitioning"**.
2. Repeatedly carry out a short simulation (approximately 100 time steps) and improve the partioning on basis of the actual computing times.
    a. Carry out a simulation using the current partitioning.
    b. Re-start Visipart, read the current partitioning and reenter the number of layers for each subdomain. This step is not necessary if Matlab is still running.
    c. Read the actual CPU-times from the **waqpro** message file using **"Options | Show Observed Runtimes"**. If the observed runtimes seem reliable, press **Accept Speeds**.
    d. Move computational grid points from one subdomain to another, by selecting them with the mouse. Keep moving points until the estimated workloads are as equal as possible.
    e. Save the partitioning using **"File | Save partitioning"**.

Steps 1(a-f) and 2(a-e) provide the best standard procedure known for the optimisation of partitionings. The following difficulties in the procedure are known:

1. The assumptions made for the estimations of processor speeds on basis of run times. The simulation calculations consist of several parts (flow, transport, turbulence), each with different work load distributions. It is possible that the processor with the heaviest work load is the restricting factor in none of these steps.

2. The assumption that the processor speed does not change when the partitioning is changed. Sometimes, very small changes in the partitioning may have very large effects on the calculation speed, mostly due to cache-effects. For example, some computers work very slowly on arrays whose size is a multiple of 256.

3. The procedure has to be restarted when the number of processors per domain (i.e. the part of the global domain in which a certain number of layers has been selected) is changed.

Refinements of the optimisation procedure and the support by Visipart may become necessary in the future.

Chapter 4

Example for assigning processes in DDHOR simulation using Visipart

In this example we illustrate the use of Visipart for DDHORsimulations.

The first step for the user is to prepare the simulation input files for the different domains.

Next, so-called areas-files must be devised for establishing the DDHOR coupling of the domains.

- parts of the domain may be rendered "inactive" (-1),
- the inactive areas must have matching openings or DDHORinterfaces.

This may be done using Visipart.

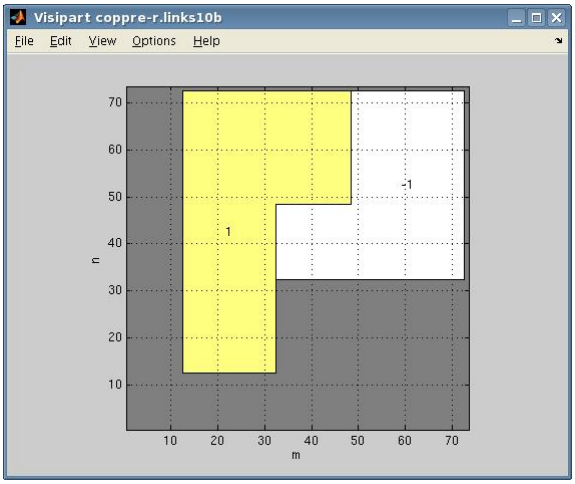
Thirdly a DDHOR config-file is needed and an initial DDHORsimulation may be performed, in which each DDHOR-domain uses a single subdomain. The way this file must be constructed is described in Section 2.7 of the document about parallel computing and domain decomposition (ugcouple.pdf). For Visipart this configuration file is parsed by Matlab. All keywords should be specified on separate lines in order for Matlab to be able to do this.

In this example we assume we have an initial partitioning for our DDHOR simulation containing three domains ("links", "rebo" and "reon"). So we need at least 3 processors to run this simulation. Each domain has a different number of layers. These 3 domains are visualised in Figure 4.1

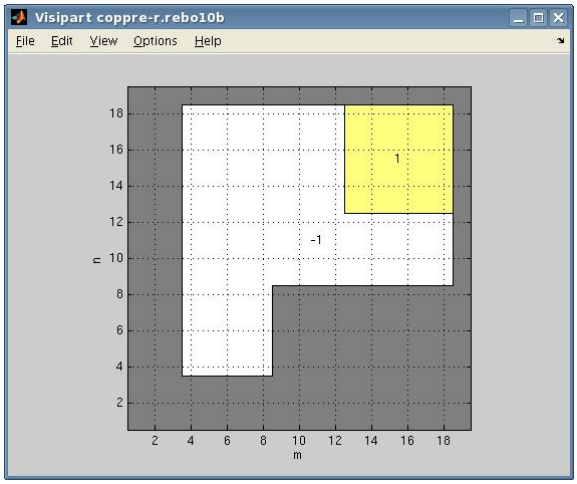
To be able to assign processors to the domains we need the run times of each domain. Note that when editing in one of the domains, Visipart does not know about the other domains. Normally a guess can be made from the number of grid points in each (sub)domain, but in case of DDHOR it is not possible to determine the number of grid points in subdomains outside your own DDHOR-domain. To read these run times (and show them) we use **"Options | Show Observed Run-Times"** and select the waqpro-m message file. We end up with Figure 4.2.

The **Accept speeds** button can now be used in the same way as for a parallel simulation, except for the fact that Visipart does not know which domain is processed currently. So if the **Accept speeds** button is used and we have a DDHOR simulation, Visipart will ask which domain is processed (see Figure 4.3).

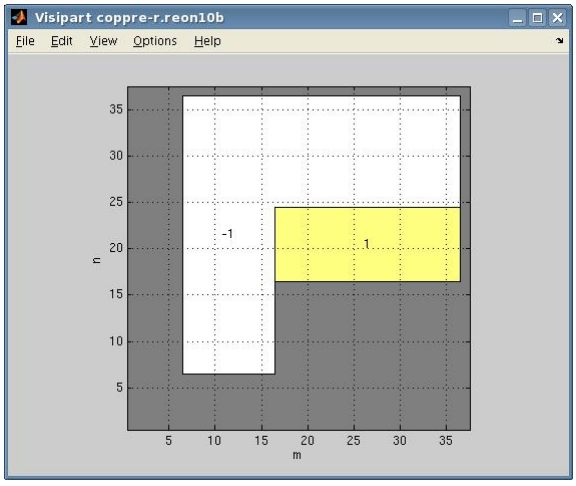
In this example we are editing the first domain and select the name "links".



(a) domain links



(b) domain rebo



(c) domain reon

Figure 4.1: The 3 domains of the example

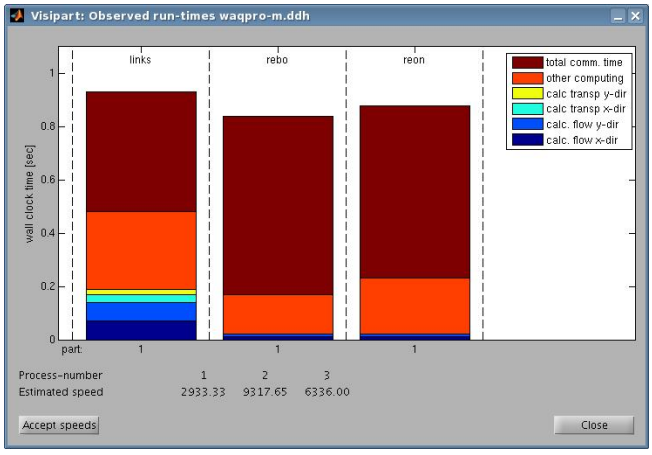


Figure 4.2: The observed runtimes of the example

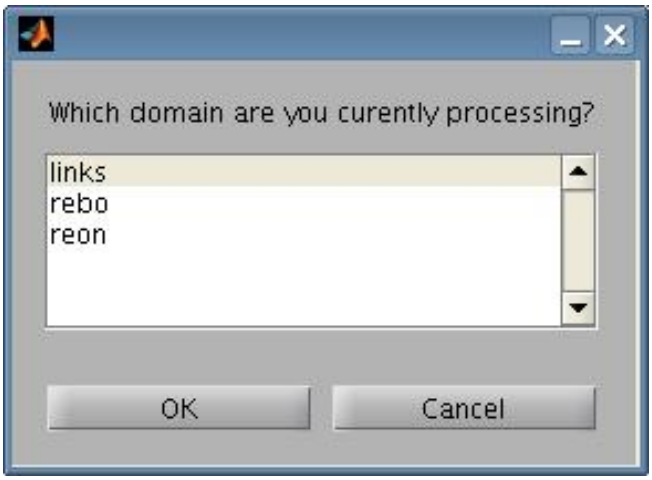


Figure 4.3: Question dialog display to select the current domain

Now that the run times for all of the domain are known, we can select **"Options | Assign Processes to Subdomains"**. This will work now in largely the same way as for a parallel run. The next figure shows an optimal assignment for our example. In our example we have 16 processors to work with, but we do not use them all if the effect on the computing time is minimal (see Figure 4.4).

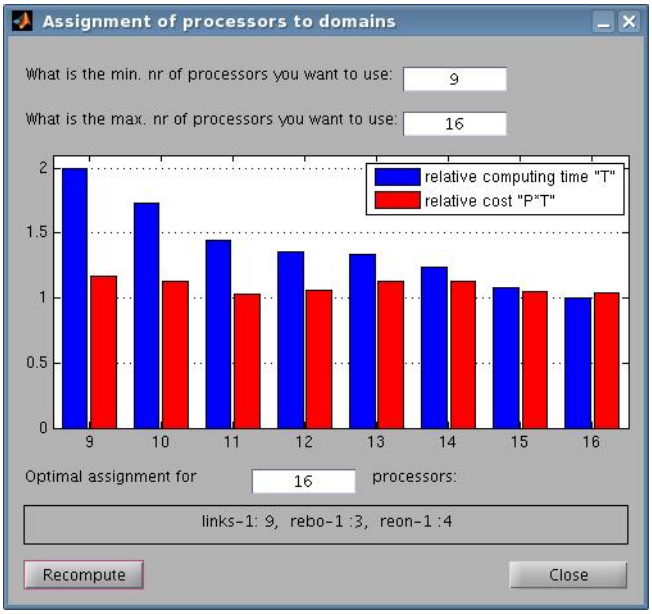


Figure 4.4: Overview of how to assign processes over the domains

From this figure we see that 9 of the processors should be used for domain "links", 3 for "rebo" and 4 for "reon".

Now that the optimal assignment is known, each domain can be split by using **"Edit | Split subdomains"**. In case of a DDHOR-simulation this must be done for each domain separately. So first we load the first partition ('links')and split it into 9 subdomains. We end up with the assignment in Figure 4.5.

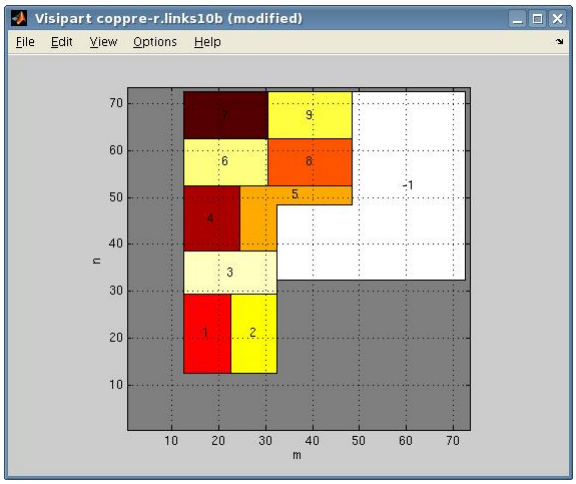


Figure 4.5: Domain 'links' is now divided in 9 subdomains

Next we load the second partition ('rebo') and split it into 3 subdomains. We end up with Figure 4.6.

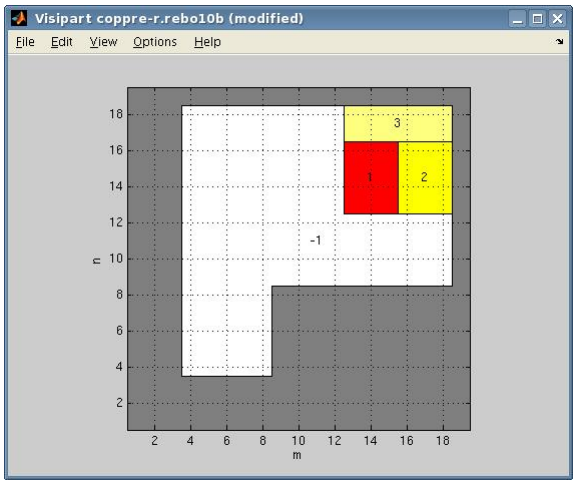


Figure 4.6: Domain 'rebo' is now divided in 3 subdomains

And finally we open the third partition ('reon') and split it into 4 subdomains, so we end with Figure 4.7.

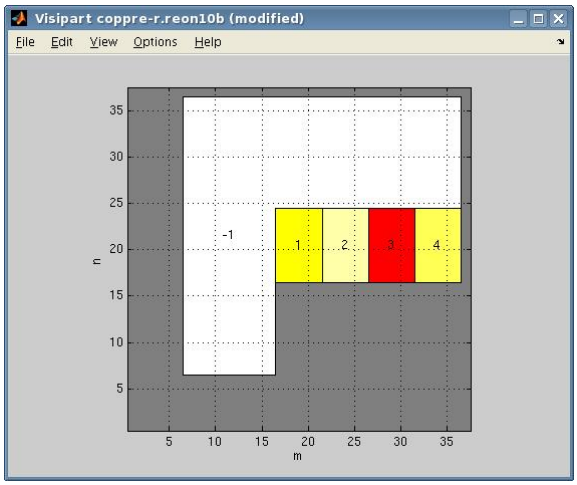


Figure 4.7: Domain 'reon' is now divided in 4 subdomains